Amendments to the Specification:

Please replace the paragraph beginning on line 13 of page 1 with the following amended paragraph:

This application is related to commonly-assigned U.S. Application Serial No. [[____]] 09/586,245, filed on even date herewith, by Mark Skryznski and Huy Ton, issued on February 10, 2004 as U.S. Patent 6,691,302, and entitled "Interfacing a Service Component to a Native API," which is incorporated herein by reference.

Please replace the paragraph beginning on line 18 of page 4 with the following amended paragraph:

In particular, specific embodiments are discussed further below. Referring to FIG. 1, a packet switched network 10 (e.g. a HiNet™ RC 3000 ToL system, which is available from Siemens Information and Communication Networks, Inc.) includes client terminals 12, 14, a multi-point control unit 16, a gateway 18, a gatekeeper 20 and a server 22. Each of the systems 12-22 is connected to a local area network (LAN) 24. Data, voice and video information may be transmitted over network 10 in accordance with connection-oriented network protocols (e.g., Transmission Control Protocol/Internet Protocol (TCP/IP)) or connectionless protocols (e.g., Ethernet, Internetwork Packet Exchange (IPX), and User Datagram Protocol (UDP)). Packet switched network 10 preferably operates in compliance with international standards for multimedia call control and application sharing (e.g., Session Initiation Protocol (SIP) [[SIP]] or H.323 for call control and T.128 for application sharing).

Please replace the paragraph beginning on line 10 of page 5 extending to page 6 with the following amended paragraph:

As shown in FIG. 2, network 10 includes a distributed dependability system that, according to a specific embodiment of the invention, may be represented as a hierarchical arrangement of nodes. One or more nodes may be contained in each of the physical systems 12-22 of network 10. The dependability system monitors the status of one or more components at each of a several different hierarchical levels 30,

32 and 34 of network 10. At network level 30, for example, the dependability system receives status information from system level 32, determines the state of "health" of the overall network 10 and takes corrective action (if necessary) based upon a set of programmed network level policies (e.g., policies relating to load sharing and system availability policies). At system level 32, the dependability system receives status information from each system component, determines the state of health of the assigned system and responds to that determination based upon a set of policies appropriate for the assigned system. Each system of network 10 may be stratified further into a component level 34, which corresponds to a logical hierarchy of system hardware, software and firmware resources. For example, gatekeeper 20 [[18]] may include a system level fault analyzer object for determining the state of health of the gatekeeper system and one or more component level fault analyzers for other components (e.g., a central processing unit (CPU), a standby CPU, an input/output (I/O) card, and one or more software components) operating inside the gatekeeper system. Many of the nodes of network 10 and many of the connections between the nodes may be duplicated. Such redundancy would improve the reliability and fault tolerance of each system and the dependability of the overall network.

Please replace the paragraph beginning on line 18 of page 6 with the following amended paragraph:

Referring to FIG. 3, according to some specific embodiments, each node of the dependability system includes a fault analyzer object 40. Fault analyzer object 40 receives status information 42 relating to the availability of one or more objects (e.g., component objects or fault analyzer objects, or both) for which fault analyzer object 40 has responsibility. Fault analyzer object 40 analyzes this information 42 and, based upon a set of programmable policies 44, issues one or more component object control instructions 46 for the objects assigned to fault analyzer object 40, and issues one or more fault analyzer object control instructions 47 for lower-level fault analyzer objects (if any) shown in FIG. 4. For example, if the current state of a critical component indicates that the component is hung, fault analyzer object 40 may issue component object control instructions to deactivate the hung component and to activate a standby

dormant component with similar functionality. Fault analyzer object 40 also may issue one or more status reports 48 (e.g., error or warning messages) for an assigned higher-level fault analyzer object.

Please replace the paragraph beginning on line 27 of page 8 extending to page 9 with the following amended paragraph:

In one embodiment, the dependability system is operable under the Windows NT operating system and is implemented by a plurality of Windows NT services written in a variety of different programming languages. The different service modules may interface with the native Windows NT operating system through a generic interface that is described in commonly-assigned U.S. Application Serial No. [[ ]] 09/586,245,filed on even date herewith, by Mark Skrzynksi and Huy Ton, issued on February 10, 2004 as U.S. Patent 6,691,302, and entitled "Interfacing a Service Component to a Native API," which is incorporated herein by reference.